

Amendments to the Claims

A complete list of pending claims follows, with indicated amendments:

1. (Currently Amended) A storage medium containing software for manipulating computer-implemented objects in a distributed system, the software comprising:

code to create a shared environment, the shared environment comprising an object-oriented programming environment distributed across multiple computer systems and comprising a plurality of objects; and

code to create an object, the object exposed to other objects in the shared environment, the object comprising:

a set of behavior ~~Behavior~~ logics, each member of the set of behavior ~~Behavior~~ logics operable ~~adapted~~ to cause the object to perform a task; and

a receiver ~~first Behavior~~ logic operable ~~, adapted~~ to receive a command ~~Command~~ from another object in the shared environment, wherein the receiver ~~first Behavior~~ logic is externally ~~invokable external to the object, the first Behavior logic comprising;~~ and

a mapping logic able to map a command received at the receiver logic, on the basis of a characteristic of the command, to a selected behavior logic for execution of the selected behavior logic

~~code to receive the Command;~~

~~code to select a Behavior logic of the set of Behavior logics corresponding to the Command from a Command Behavior mapping; and~~

~~code to execute the selected Behavior logic responsive to the Command.~~

2. (Currently Amended) The storage medium of claim 1, wherein the set of behavior Behavior logics and the mapping logic are ~~Command-Behavior-mapping~~ private to the object.

3. (Currently Amended) The storage medium of claim 1, wherein the set of behavior Behavior logics has ~~having~~ no members.

4. (Currently Amended) The storage medium of claim 1, the object further comprising:

a default behavior Behavior logic, ~~adapted~~ operable to cause the object to perform a default task, wherein the default behavior Behavior logic is private to the object and wherein the default behavior logic is executed if the received command is not mapped to another behavior logic;

~~the first Behavior logic further comprising:~~

~~code to execute the default Behavior logic responsive to the Command if no Behavior logic is selected by the code to select a Behavior logic corresponding to the Command.~~

5. (Currently Amended) The storage medium of claim 1, wherein a command can be mapped to multiple behavior logics ~~the Command-Behavior-mapping can cause the code to select a Behavior to select multiple Behaviors.~~

6. (Currently Amended) The storage medium of claim 1, the object further comprising:

an authentication data, the authentication data providable to other objects for authenticating commands ~~Commands~~ received from the other objects by the code to receive the command ~~Command~~.

7. (Currently Amended) The storage medium of claim 6, wherein the command ~~Command~~ comprises the authentication data, and wherein the mapping of a command to a behavior logic may be restricted in response ~~the Command-Behavior mapping restrictable responsive~~ to the authentication data.

8. (Currently Amended) The storage medium of claim 1, the software further comprising:

code to create a first Shadow of the object, the first Shadow of the object operable ~~adapted~~ to communicate with the object, the first Shadow of the object being informed of changes to the object and the object being informed of changes to the first Shadow of the object.

9. (Original) The storage medium of claim 8, wherein the first Shadow of the object is a copy of the object.

10. (Currently Amended) The storage medium of claim 8, wherein the mapping of commands to behavior logics ~~Command-Behavior-mapping~~ of the first Shadow of the object

differs from the mapping of commands to behavior logics ~~Command-Behavior mapping~~ of the object.

11. (Currently Amended) The storage medium of claim 8, the software further comprising:

code to create a plurality of Shadows of the object operable ~~adapted~~ to communicate with the object and the first Shadow of the object, the object and the first Shadow of the object being informed of changes to any of the plurality of Shadows of the object and each of the plurality of Shadows of the object being informed of changes to the object and changes to the first Shadow of the object.

12. (Original) The storage medium of claim 8, the software further comprising:
code to promote the first Shadow of the object into a new object.

13. (Original) The storage medium of claim 12, the software further comprising:
code to create a plurality of Shadows of the object,
wherein executing the code to promote the first Shadow of the object into a new object converts each of the plurality of Shadows of the object into a Shadow of the new object.

14. (Currently Amended) The storage medium of claim 12, the shared environment further comprising:

a plurality of computer systems servers, the object on a first computer system server of the plurality of computer systems servers, the first Shadow of the object on a second computer system server of the plurality of computer systems servers; and

code to manage the plurality of computer systems servers, executing the code to promote the first Shadow of the object to a new object if the first computer system server experiences a predetermined condition.

15. (Currently Amended) The storage medium of claim 1, the set of Behavior logics further comprising:

code to modify the mapping logic ~~Command-Behavior mapping to cause the to~~
~~modify the mapping of commands to behavior logics~~ ~~code to select a Behavior logic responsive~~
~~to the Command to select a different Behavior logic of the set of Behavior logics.~~

16. (Currently Amended) The storage medium of claim 1,
wherein the shared environment is operable to execute on ~~comprising:~~ a plurality
of computer systems servers; and

wherein the object has ~~having~~ a location on one of the plurality of computer
systems servers; and wherein the object ~~acting independent~~ acts independently of the its location.

17. (Currently Amended) The storage medium of claim 1, the object further comprising:

code to configure the mapping logic ~~create the Command-Behavior mapping~~ from
an external data source.

18. (Currently Amended) The storage medium of claim 1, wherein the software is capable of using any networking protocol.

19. (Currently Amended) A method of manipulating a computer-implemented object in a distributed system, the method comprising the steps of:

creating a shared environment;³ the shared environment comprising an object-oriented programming environment distributed across multiple computer systems and comprising a plurality of objects; and

creating an object, wherein the object is exposed to other objects in the shared environment, and wherein the step of creating an object comprises ~~comprising~~ the step of:

coding a set of behavior ~~Behavior~~ logics, each member of the set of behavior ~~Behavior~~ logics causing the object to perform a task;

manipulating the object, wherein the step of manipulating the object comprises ~~comprising~~ the steps of:

receiving a command ~~Command~~ from another object of the plurality of objects in the shared environment;

selecting a behavior ~~Behavior~~ logic of the set of behavior ~~Behavior~~ logics corresponding to the command ~~Command~~ on the basis of a mapping logic within the object that maps commands to behavior logics of the set of behavior logics on the basis of a characteristic of the command ~~from a Command-Behavior mapping~~; and

executing the selected behavior ~~Behavior~~ logic responsive to the command ~~Command~~.

20. (Currently Amended) The method of claim 19, wherein the set of behavior Behavior logics and the ~~Command-Behavior mapping~~ logic are private to the object.

21. (Currently Amended) The method of claim 19, further comprising the step of:
modifying the mapping logic to modify the mapping of commands to behavior
logics changing the Command-Behavior mapping, causing the step of selecting a Behavior logic
to select a different Behavior logic of the set of Behavior logics responsive to the Command.

22. (Currently Amended) The method of claim 19, the method further comprising the steps of:

coding a default behavior Behavior logic to cause the object to perform a default task;; and

executing the default behavior Behavior logic if no other behavior Behavior logic from the set of behavior logics is mapped to the received command ~~selected by the step of selecting a Behavior logic.~~

23. (Currently Amended) The method of claim 19, wherein the set of behavior Behavior logics has ~~having~~ no members.

24. (Currently Amended) The method of claim 19, wherein multiple behavior logics are mapped to and selected on the basis of a received command ~~the Command-Behavior mapping~~
~~can cause the step of selecting a Behavior logic to select multiple Behaviors.~~

25. (Currently Amended) The method of claim 19, further comprising the step steps of:

creating an authentication data for the object.

26. (Currently Amended) The method of claim 25, the command ~~Command~~ comprising the authentication data, the method further comprising the step of:

restricting the mapping of commands to behavior logics in response ~~Command-Behavior mapping~~ responsive to the authentication data.

27. (Currently Amended) The method of claim 19, further comprising the step of:
creating a first Shadow of the object, the first Shadow of the object operable adapted to communicate with the object, the first Shadow of the object being informed of changes to the object and the object being informed of changes to the first Shadow of the object.

28. (Original) The method of claim 27, the step of creating the first Shadow of the object comprising the step of:

copying the object.

29. (Currently Amended) The method of claim 27, the step of creating the first Shadow of the object comprising the step of:

modifying the mapping of commands to behavior logics ~~Command-Behavior~~ logic of the first Shadow of the object.

30. (Currently Amended) The method of claim 27, further comprising the step of:
creating a plurality of Shadows of the object; operable ~~adapted~~ to communicate with the object and the first Shadow of the object, the object and the first Shadow of the object being informed of changes to any of the plurality of Shadows of the object and each of the plurality of Shadows of the object being informed of changes to the object and changes to the first Shadow of the object.

31. (Original) The method of claim 27, further comprising the step of:
promoting the first Shadow of the object into a new object.

32. (Original) The method of claim 31, further comprising the step of:
creating a plurality of Shadows of the object,
converting each of the plurality of Shadows of the object into a Shadow of the new object, responsive to the step of promoting the first Shadow of the object.

33. (Currently Amended) The method of claim 19, ~~the shared environment~~
~~comprising:~~
~~a plurality of servers;~~
wherein the object has a location on a first computer system ~~server~~ of the plurality of computer systems ~~servers~~; and wherein the object acting independent acts independently of the
its location.

34. (Original) The method of claim 19, the shared environment capable of using any networking protocol to communicate with another shared environment.

35. (Currently Amended) The method of claim 19, further comprising the step of:
creating the ~~Command-Behavior~~ mapping logic from an external data source.

36. (Currently Amended) A method for developing ~~of designing~~ an application for execution on at least one computer system from configurable objects having behavior ~~Behavior~~ logics capable of performing tasks, the method comprising the steps of:

defining within an object-oriented programming environment ~~creating~~ a plurality of objects, each object of the plurality of objects operable ~~adapted~~ to receive and execute commands ~~Commands~~, each object exposed to each other object of the plurality of objects, the step of creating the plurality of objects comprising the steps of:

creating a set of behavior ~~Behavior~~ logics for an object, ~~the set of Behavior logics capable of being an empty set;~~

mapping members of a first set of commands ~~Commands~~ to members of the set of behavior ~~Behavior~~ logics, wherein the mapping function of an object is included within the object;

mapping any command ~~Command~~ not a member of the first set of commands ~~Commands~~ to a default behavior ~~Behavior~~ logic; and

configuring a receiver ~~Command-receiver~~ Behavior logic to receive a command ~~Command~~ and initiate the execution of ~~execute the~~ a behavior ~~Behavior~~ logic

corresponding to the command in response to the mapping of the command to the behavior logic Command.

37. (Currently Amended) The method of claim 36, further comprising the steps of:
creating a Shadow of an object of the plurality of objects, the Shadow configured such that sending a command Command to the Shadow causes the object to act as if the command Command had been sent to the object.

38. (Currently Amended) The method of claim 37, each of the plurality of objects having a location on one of a plurality of computer systems servers, each of the plurality of objects being independent of the location of each other of the plurality of objects.

39. (Currently Amended) The method of claim 38, wherein a Shadow of each of the plurality of objects is automatically created on each of the plurality of servers other than the server on which the object is located.

40. (Currently Amended) A processor-based system, comprising:
a first processor; and
a first storage device coupled to the first processor containing a software to manipulate computer-implemented objects in a shared environment, the software comprising:
code to create a shared environment, the shared environment comprising an object-oriented programming environment distributed across multiple computer systems and comprising a plurality of objects; and

code to create an object of the plurality of objects, the object exposed to other objects in the shared environment, the object comprising:

a set of behavior ~~Behavior~~ logics, each member of the set of behavior ~~Behavior~~ logics operable ~~adapted~~ to cause the object to perform a task; and

a receiver ~~first-Behavior~~ logic; operable ~~adapted~~ to receive a command ~~Command~~ from another object in the shared environment, the receiver ~~first-Behavior~~ logic being externally invokable external to the object, the first Behavior logic comprising;

a mapping logic able to map a command received at the receiver logic to a selected behavior logic for execution of the selected behavior logic on the basis of a characteristic of the command

~~code to receive the Command;~~

~~code to select a Behavior logic of the set of Behavior logics corresponding to the Command from a Command Behavior mapping; and~~

~~code to execute the selected Behavior logic responsive to the Command.~~

41. (Currently Amended) The processor-based system of claim 40, the object further comprising:

a default behavior ~~Behavior~~ logic; adapted ~~operable~~ to cause the object to perform a default task, wherein the default Behavior logic is private to the object and wherein the default behavior logic is executed if the received command is not mapped to another behavior logic;
~~the first Behavior logic further comprising;~~

~~code to execute the default Behavior logic responsive to the Command if no Behavior logic is selected by the code to select a Behavior logic corresponding to the Command.~~

42. (Currently Amended) The processor-based system of claim 40, wherein a command can be mapped to multiple behavior logics ~~the Command-Behavior mapping can cause the code to select a Behavior logic to select multiple Behaviors.~~

43. (Currently Amended) The processor-based system of claim 40, further comprising:

an input device coupled to the first processor,

wherein a first object of the plurality of objects is coupled to the input device such that manipulation of the input device sends a command ~~Command~~ from the first object to a second object of the plurality of objects without identifying the input device, the second object of the plurality of objects acting responsive to the command ~~Command~~ independent of the nature of the input device.

44. (Original) The processor-based system of claim 40, further comprising:

an output device coupled to the first processor,

wherein a first object of the plurality of objects is coupled to the input device such that a first object is capable of rendering a second object on the output device without identifying the output device to the second object.

45. (Original) The processor-based system of claim 40, further comprising:
a second processor;
a network, coupled to the first processor and the second processor;
a second storage device coupled to the second processor, the second storage device containing the software;
the software further comprising:
code to connect the shared environment to the network;
code to create a Shadow on the second processor of the object on the first processor, the Shadow and the object communicating with each other to inform the Shadow of changes to the object and the object of changes to the Shadow.

46. (Currently Amended) A software architecture for manipulating computer-implemented objects on a plurality of computers, some of the plurality of computers having input devices and some of the plurality of computers having output devices, the software architecture implemented in an extensible object-oriented language, comprising:

a distributed system, comprising:
a plurality of shared environments, each of the plurality of shared environments comprising an object-oriented programming environment distributed across and executing on a different computer of the plurality of computers, each of the plurality of shared environments comprising:

a CommandReceiver class, the CommandReceiver class comprising:

a set of Behavior private methods, each member of the set of Behavior methods operable ~~adapted~~ to cause instantiations of the CommandReceiver class to perform a task; and

an executeCommand public method, operable ~~adapted~~ to receive a Command from an object in the shared environment, the executeCommand public method comprising:

code to receive the Command;

code to select a Behavior private method of the set of Behavior private methods selected corresponding to a characteristic of the Command from a Command-Behavior mapping; and

code to execute the selected Behavior private method; and

a Kernel subclass of the CommandReceiver class, the Kernel class comprising:

code to instantiate objects of the CommandReceiver class;

code to destroy objects of the CommandReceiver class.

47. (Currently Amended) The software architecture of claim 46, further comprising:

a Pawn subclass of the CommandReceiver class, the Pawn subclass comprising:

code to register an instantiation of a Pawn with a Kernel object of the Kernel subclass;

code to determine whether an instantiation ~~instantion~~ of the Pawn subclass is a real Pawn or a Shadow Pawn of a real Pawn, and

code to send State information about an instantiation of the Pawn subclass,
wherein Commands received by Shadow Pawns are sent to the real Pawn
corresponding to the Shadow Pawn.

48. (Original) The software architecture of claim 46, further comprising:
a ControlDevice subclass of the CommandReceiver class corresponding to an
input device for receiving input from the input device and sending Commands to other
CommandReceiver objects.

49. (Original) The software architecture of claim 46, further comprising:
a Construct subclass of the CommandReceiver class corresponding to an output
device for rendering objects of the CommandReceiver class with graphical attributes.

50. (Original) The software architecture of claim 46, further comprising:
a Console subclass of the CommandReceiver class for allowing a user of the
distributed system to instantiate, modify, and destroy objects, and for allowing a user to send
Commands to CommandReceiver objects.

51. (Original) The software architecture of claim 46, further comprising:
a Nengine subclass of the CommandReceiver class for serializing and
deserializing CommandReceiver objects, transmitting and receiving the serialized
CommandReceiver object across a network to a Nengine in another shared environment of the
distributed system.

52. (Original) The software architecture of claim 51, further comprising:

a Node subclass of the CommandReceiver class, an instantiation of the Node subclass corresponding to a Pawn object for representing the Pawn object to a Nengine object for communicating State information corresponding to a Pawn to Shadow Pawns of the Pawn and for communicating Commands sent to a Shadow Pawn to the real Pawn corresponding to the Shadow Pawn.